

# IEEE Standard 1500 Compatible Interconnect Diagnosis for Delay and Crosstalk Faults

Katherine Shu-Min Li, *Member, IEEE*, Chauchin Su, *Member, IEEE*, Yao-Wen Chang, *Member, IEEE*, Chung-Len Lee, *Member, IEEE*, and Jwu E. Chen, *Member, IEEE*

**Abstract**—An interconnect diagnosis scheme based on the oscillation ring (OR) test methodology for systems-on-chip (SOC) design with heterogeneous cores is proposed. In addition to traditional stuck-at and open faults, the OR test can also detect and diagnose important interconnect faults such as delay faults and crosstalk glitches. The large number of test rings in the SOC design, however, significantly complicates the interconnect diagnosis problem. In this paper, the diagnosability of an interconnect structure is first analyzed then a fast diagnosability checking algorithm and an efficient diagnosis ring generation algorithm are proposed. It is shown in this paper that the generation algorithm achieves the maximum diagnosability for any interconnect. Two optimization techniques are also proposed, an adaptive and a concurrent diagnosis method, to improve the efficiency and effectiveness of interconnect diagnosis. Experiments on the MCNC benchmark circuits show the effectiveness of the proposed diagnosis algorithms. In all experiments, the method achieves 100% fault detection coverage and the optimal interconnect diagnosis resolution.

**Index Terms**—Crosstalk fault, delay fault, fault diagnosis, interconnections, oscillation ring (OR) test scheme.

## I. INTRODUCTION

INTERCONNECT delays, rather than gate delays, dominate overall circuit performance in the nanometer era [1]–[3], especially for systems-on-chip (SOC) designs. The SOC design methodology has become a reality in IC industry. Integrating reusable cores from multiple sources is essential in SOC designs, and different design-for-testability methodologies are usually required to test different cores. In particular, the long and parallel global interconnects incur significant delay and crosstalk glitch faults, and thus, special interconnect detection and diagnosis schemes are desirable.

Manuscript received May 5, 2005; revised August 8, 2005 and October 20, 2005. The work of C. Su was supported in part by the National Science Council of Taiwan under Grants NSC 95-2221-E-009-334 and NSC 95-2221-E-009-328-MY3. The work of Y.-W. Chang was supported in part by the National Science Council of Taiwan under Grants NSC 93-2815-C-002-046-E, NSC 94-2215-E-002-005, and NSC 94-2752-E-002-008-PAE. This paper was recommended by Associate Editor R. D. Blanton.

K. S.-M. Li is with the Department of Computer Science and Engineering, National Sun Yat-sen University, Kaohsiung 804, Taiwan, R.O.C.

C. Su is with the Department of Electrical and Control Engineering, National Chiao Tung University, Hsinchu 300, Taiwan, R.O.C.

Y.-W. Chang is with the Department of Electrical Engineering and Graduate Institute of Electronics Engineering, National Taiwan University, Taipei 106, Taiwan, R.O.C.

C.-L. Lee is with the Department of Electronics Engineering, National Chiao Tung University, Hsinchu 300, Taiwan, R.O.C.

J. E. Chen is with the Department of Electrical Engineering, National Central University, Chungli 32001, Taiwan, R.O.C.

Digital Object Identifier 10.1109/TCAD.2006.881330

Interconnect test and diagnosis for various applications, such as printed circuit board, multichip module, and systems in package, have been studied extensively in the literature [4]–[19]. Previous works on interconnect test, including fault detection and diagnosis, focus mainly on traditional fault models, including stuck-at and bridging faults. Those diagnosis algorithms include counting sequence, walking-0 and walking-1 sequence, maximal independent test set [4]–[12], etc. An efficient way to apply these tests is to exploit the boundary-scan architecture [13], [14]. Many diagnosis algorithms presented in previous works focus mainly on special interconnect structures, especially for bus-oriented systems [15], sparsely interconnected systems [16], or field programmable gate array (FPGA) designs [17]–[19]. The diagnosis of wire delay and crosstalk faults, often considered the most important segments of interconnect diagnosis, has attracted increasing attention since the process technology enters the deep submicrometer era. Much work has been done in these areas, including the development of fault models, test generation algorithms, and test methodology for delay tests [14] and built-in self test (BIST) schemes for crosstalk faults [20]–[22].

Oscillation ring (OR)-based test is an efficient and effective method to detect faults in a circuit or a device [23], [24]. An OR is a closed loop with an odd number of signal inversions. Once the ring is constructed, oscillation signal appears on the ring. For a circuit with faults, some rings will not oscillate correctly. Once a set of oscillation tests has been conducted, we can locate some or all of the faults according to the test outcome [25]. Whether each fault can be correctly identified or diagnosed depends on the interconnect structure and the test rings applied. The advantage of applying OR-based diagnosis for the interconnect structure [25] is that, in addition to functional faults like stuck-at and open faults, it is also capable of identifying delay and crosstalk glitch faults, which are the main sources for the loss of signal integrity [20]–[22]. Therefore, the OR-based technique is an ideal approach to interconnect diagnosis.

In this paper, we propose an OR-based scheme to diagnose interconnect faults for SOC designs. Unlike previous works, our scheme has the following features: 1) It is applicable to the general interconnect structure; 2) it is compatible with the IEEE standard 1500 core test standard for SOC [26]–[28] by providing enhanced IEEE standard 1500-compliant wrapper designs; and 3) in addition to traditional fault models (stuck-at and open faults), delay and crosstalk glitch faults can also be handled with this approach.

The major challenge for diagnosis is the large number of test rings to be processed, as the number of rings can be exponential to the number of nets and usually very large in an SOC. Therefore, it is very time consuming and often impossible to find out a minimum set of test rings to achieve the maximal diagnosability. We propose an efficient ring generation algorithm for OR-test-based interconnect diagnosis. We summarize our main contributions as follows.

- 1) We give a theoretic analysis of the diagnosability for any general interconnect structure and propose a fast diagnosability checking algorithm, which greatly reduces the time complexity.
- 2) We propose an efficient ring generation algorithm for interconnect diagnosis, and the goal of this algorithm is to minimize the number of required test rings. It exploits the fast diagnosability check to accelerate the ring generation process.
- 3) We propose two optimization techniques to further improve the test time: 1) An adaptive diagnosis method dynamically applies the next test diagnosis ring according to the result of the previous test, and it reduces the test time by 1.54X–2.67X compared to the predetermined diagnosis method. 2) A concurrent diagnosis method, in which multiple compatible rings are applied simultaneously, improves test effectiveness by up to 9.66%. In particular, neither of the two techniques incurs any hardware overhead.

Experiments on the Microelectronics Center for North Carolina (MCNC) benchmark circuits show the effectiveness of the proposed diagnosis algorithm. In all experiments, our method achieves 100% fault coverage and the optimal diagnosis resolution. Here, the diagnosis resolution is defined as the maximum number of nets with the same syndrome under a given set of test diagnosis rings and the single-fault assumption, and the optimal diagnosis resolution is defined as a test that can diagnose all interconnects and the maximum number is one.

The remainder of this paper is organized as follows. Section II presents some preliminary information on the OR-test scheme for the interconnect detection and diagnosis and gives the problem formulation for the problem addressed in this paper. In Section III, we first analyze the diagnosability of an interconnect structure and then present an algorithm for quick diagnosability check. A theoretical analysis of the lower and upper bounds for the interconnect detection and diagnosis test scheme is also given. Section IV presents an efficient integrated interconnect diagnosis algorithm that includes both a fast diagnosability check heuristic and a diagnosis ring generation algorithm. Section V presents two optimization mechanisms for interconnect diagnosis. Experimental results are reported in Section VI, and finally, a brief conclusion is given in Section VII.

## II. OR FOR INTERCONNECT TESTING AND DIAGNOSIS

In this section, we give some preliminary knowledge of the OR-test scheme for interconnect test, including the global test structure, basic test operations, the frequency measurement

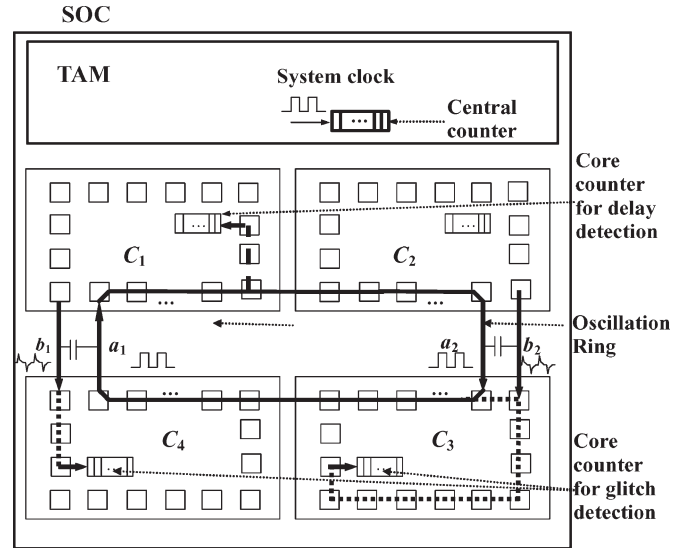


Fig. 1. Test architecture for wire delay and crosstalk detection and wire delay measurement.

formula, modified wrapper cell designs, and the interconnect model for diagnosis. In the last part, we give the motivation of this research and formally formulate the problem.

### A. OR-Test Architecture

In this section, we discuss the interconnect OR test (IORT) for SOC interconnects [25]. Fig. 1 illustrates the global counter-based test architecture for both wire delay and crosstalk glitch detection for SOC IC interconnects. This test architecture implements the IEEE standard 1500-compliant core test standard. In IEEE standard 1500, each input/output pin of a core is attached with a wrapper cell, and a centralized test access mechanism (TAM) is provided to coordinate all test processes. In addition to the normal input/output connections, all wrapper cells in a core can also be connected with a shift register, usually referred to as a scan path, to facilitate test access. An enhanced wrapper cell design has been proposed to provide extra connections and inversion control so that the ORs can be constructed through the wires and the boundary-scan paths in cores [25]. For example, the OR-test architecture shown in Fig. 1 is consist of one OR and two neighboring nets, and the scan paths in cores C1, C2, C3, and C4 are part of the OR.

The target fault models of this test architecture are stuck-at, open, wire delay, and crosstalk glitch faults. In addition to fault detection, measuring the wire delay fault can also be achieved. If an OR fails to oscillate, there exists stuck-at or open fault(s) in the components of the OR. The period of the oscillation signal is measured by using a delay counter in a core to test wire delay faults, and a similar scheme is also applied for crosstalk glitch detection.

A local counter is included in each core, and a central counter is in the TAM of an SOC. The central counter in the TAM is enabled by the signal *OscTest* and triggered by the system clock. A local counter is connected to one wrapper cell in each core; however, it can be accessed by every wrapper cell through the wrapper cell chain. When an OR passes a core, an internal

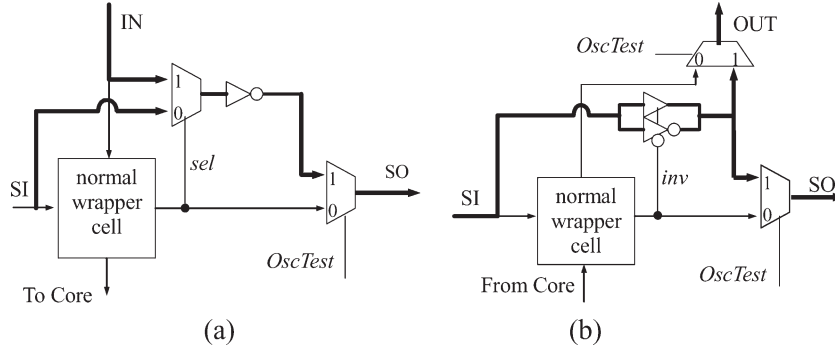


Fig. 2. Enhanced wrapper cells with forced inversion: (a) input and (b) output.

scan path is formed to connect the oscillation signal to the local counter. For example, consider core  $C_1$ , which is passed by the OR in Fig. 1. The oscillation signal is fed to the local counter through a series of modified wrapper cells. When an oscillation test session starts ( $OscTest = 1$ ), the TAM enables its own central counter as well as all local counters in the cores. After the central counter in the TAM counts to a specific number  $n$ , the oscillation test session terminates, and all local counters are disabled ( $OscTest = 0$ ). Then, all the local counter contents can then be scanned out to automatic test equipment (ATE) for inspection.

Assume that  $m$  ORs are tested. Let the frequency of the system clock be  $f$  and the delay counter contents of the rings be  $n_1, n_2, \dots, n_m$ , respectively. An estimation of the  $i$ th ring's oscillation frequency  $f_i$  can be approximated by

$$f_i = f \times n_i / n. \quad (1)$$

Since the frequency of each ring is predetermined during the design phase, a wire delay fault is detected and measured by inspecting the contents of the delay counters.

Let the oscillation frequency of the rings, according to the timing specification, be  $f_{\min} \leq f_i \leq f_{\max}$ , with the unit of measuring  $T_0 (= n/f)$ . Thus, we have  $n_{\min} \leq n_i \leq n_{\max}$ , where  $n_{\min} = f_{\min} \times T_0$  and  $n_{\max} = f_{\max} \times T_0$ . Let  $\xi$  be the resolution of delay measurement and  $\varepsilon$  be the maximum measurement error. Since a counter's maximum measurement error is  $\pm 1$ , the requirement for  $\varepsilon$  should be the reciprocal of  $f_{\min}$  and  $T_0$ .

$$\varepsilon = \frac{1}{f_{\min} \times T_0} \leq \zeta. \quad (2)$$

An example for delay measurement is given as follows. Let the frequency specification of the ORs be 4–400 MHz and  $\xi$  be 0.001, implying the counter content  $n_{\min}$  is at least 1000. From (2), we have the required  $T_0$  to be 250  $\mu$ s. This example illustrates the feasibility of the oscillation test scheme from a measurement prospect, and this frequency specification is actually compliant with ATE specifications.

In order to detect the crosstalk glitch in Fig. 1, consider wire  $b_1$  and  $b_2$  in Fig. 1, and assume that there is a coupling crosstalk effect between victim nets  $b_1$  (or  $b_2$ ) and the aggressor nets  $a_1$  (or  $a_2$ ) of the OR. Interconnects ( $b_1$  and  $b_2$ ) adjacent to an OR are affected by the oscillation signal if there is an excessive coupling capacitance between these two lines ( $a_1$  and  $b_1$ , and

$a_2$  and  $b_2$ ) [29]. When the oscillation signal occurs, crosstalk-induced glitches appear on the victim nets  $b_1$  and  $b_2$ . Similar to the case for wire delay detection, the glitches on net  $b_1$  ( $b_2$ ) are sent to local counters in core  $C_4$  ( $C_3$ ) through a series of modified wrapper cells. Since there is an inverter per modified wrapper cell in the OR-test mode, the induced glitches are amplified when the glitches pass through the wrapper cells, and the amplified glitches are used to trigger the local counters in core  $C_3$  and  $C_4$  for glitch detection and diagnosis.

### B. Enhanced IEEE Standard 1500-Compliant Wrapper Cell Design

An OR for interconnect testing is consist of interconnect wires and parts of the scan path in each core where the ring passes. Therefore, an IEEE standard 1500-compliant wrapper cell must provide necessary paths between input/output ports and scan in/out ports. If an oscillation test is used to test wires attached to/from pads, an IEEE standard 1500-compliant boundary-scan cell also has to be modified in a similar way in order to facilitate the scheme. In this section, we present the enhanced wrapper cell designs.

A normal wrapper cell provides two types of paths: a scan path connecting all wrapper cells into a shift register and an interface buffering between the internal core and the wire connected to the pin. Whenever an oscillation test is applied, a third combination path must be provided. For an input pin, the wrapper cell must connect the pin input (IN) to the scan output (SO); while for an output pin, it should connect scan in (SI) to pin output (OUT) during an oscillation test session.

The enhanced wrapper cell design is shown in Fig. 2 for the input and output cells. In each cell, two multiplexers (MUXs) are added for path selection. For an input wrapper cell, the extra paths are  $SI \rightarrow SO$  and  $IN \rightarrow SO$ ; while for an output wrapper cell, the extra paths are  $SI \rightarrow SO$  and  $SI \rightarrow OUT$ . The added inverting and noninverting buffers are used to generate oscillation signals for the OR test; however, in an input wrapper cell, only one type of buffer is provided due to the limited control signals. We assume that an inverter is used in an input cell. Two control signals are needed in each enhanced wrapper: signal  $OscTest$  is a global control signal, while signal  $sel$  is only used in the input wrapper cell and signal  $inv$  is only used in the output wrapper cell to ensure the odd parity of each ring. Signals  $sel$  and  $inv$  are set individually and scanned into the wrapper cells before an OR-test session starts.

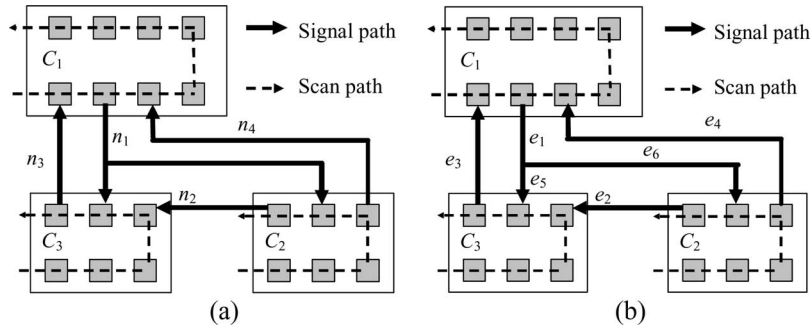


Fig. 3. Sample SOC circuit. (a) Hypergraph and four hypernets in interconnect structure. (b) Labeling all net segments or edges.

C. Interconnect Graph Models

In this section, we describe the graph model for interconnect diagnosis and the notations that will be used in the following sections.

An SOC circuit example consisting of three cores (\$C\_1\$, \$C\_2\$, and \$C\_3\$) and four nets (\$n\_1\$, \$n\_2\$, \$n\_3\$, and \$n\_4\$) is illustrated in Fig. 3(a). In an OR-based interconnect test scheme, one must construct rings to cover all wires and ensure that there is an odd number of signal inversions on every ring. In order to form the rings and generate oscillation signals, the wrapper cells need to be modified to provide extra paths and inversion control. The OR test can detect and diagnose stuck-at and open crosstalk glitch faults on nets, while wire delay faults can be measured with the help of simple built-in test architectures, as shown in Fig. 1.

For example, there are three rings in Fig. 3. The first ring consists of nets \$n\_1\$ (and its right-hand side branch), \$n\_2\$, and \$n\_3\$, and it passes all three cores. The second ring consists of \$n\_1\$ (and its left-hand side branch) and \$n\_3\$, and scan paths in \$C\_1\$ and \$C\_3\$. The third ring consists of nets \$n\_1\$ (and its right-hand side branch) and \$n\_4\$, and scan paths in \$C\_1\$ and \$C\_2\$.

Since the internal scan paths can be separately tested and diagnosed, we shall assume that they are fault free. The diagnosis problem is restricted to interconnect wires among modules in this paper. The goal for interconnect diagnosis is to diagnose any single fault for each net segment to achieve the optimal diagnosis resolution. For diagnosis purposes, all wire segments of a net are different since they may be passed by different rings. Therefore, we shall assign distinct labels to them. For example, in Fig. 3(b), the seven net segments are labeled as edges \$e\_1\$-\$e\_7\$, and our goal is to diagnose any single fault on every edge or net segment to achieve the optimal diagnosis resolution.

To perform the interconnect test for detection or diagnosis with an OR, we must find rings to cover all nets to be tested. In order to simplify the interconnect diagnosis problem, we model the SOC circuit by a hypergraph, and model interconnects by a hypernet in Fig. 4.

*Definition 1:* A hypergraph \$H = (V, L)\$ consists of a vertex set \$V\$ and an edge set \$L\$. An edge set \$L\$ is consist of multiterminal edges connecting a set of vertices \$V\_i \subseteq V\$, \$|V\_i| \ge 2\$. Such an edge is referred to as a hypernet.

For example, \$n\_1\$ in Fig. 3 is a hypernet connecting three terminals (pins). Furthermore, we assume that, in an \$n\$-terminal hypernet, one terminal is the source node (i.e., sending

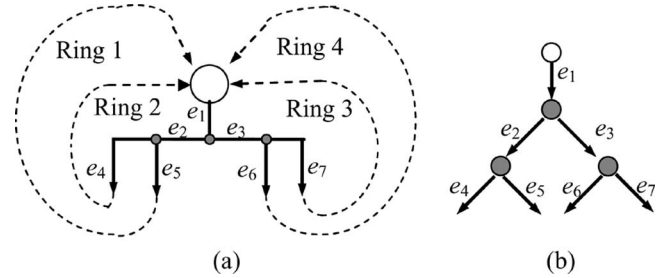


Fig. 4. (a) Hypernet. (b) Corresponding interconnect diagnosis graph model.

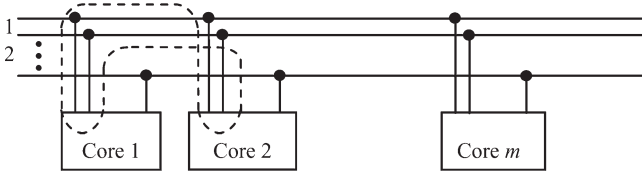
signal) while the others \$n - 1\$ are the sink nodes (i.e., receiving signals).

In Fig. 3, the circuit structure of an SOC can be directly transformed into a hypergraph, and each pin is a vertex, while each signal net is a hypernet. However, this graph model is not good enough for diagnosis, since different parts of the same net (i.e., different net segments) affect different rings. Consider the five-terminal hypernet shown in Fig. 4(a), divided into seven net segments \$e\_1\$-\$e\_7\$. If edge \$e\_1\$ is faulty, all four rings will not oscillate correctly. A faulty \$e\_2\$ affects rings 1 and 2, a faulty \$e\_3\$ affects rings 3 and 4, and faults on edges \$e\_4\$, \$e\_5\$, \$e\_6\$, and \$e\_7\$ affect rings 1, 2, 3, and 4, respectively. For diagnosis purposes, all these seven segments are different.

From the previous discussion, the hypernet cannot be used for diagnosis. Therefore, we transform an interconnect structure into an interconnect diagnosis graph model as follows. The scan path and wrapper cells in a core are lumped into a single terminal node, as we assume that they are fault-free. The fan-out points of a hypernet form dummy intermediate nodes, and a net segment connecting two nodes is modeled as an edge. For example, the graph model for the hypernet in Fig. 4(a) is transformed into an interconnect diagnosis graph model in Fig. 4(b), where the white node is a terminal node and gray nodes are intermediate nodes. An edge is the smallest unit of net segments that can be uniquely diagnosed. For diagnosis observation, any stem edge affects all its downstream nodes and edges in Fig. 4(b).

*Definition 2:* A directed graph \$G = (V, E)\$ consists of a vertex set \$V\$ and an edge set \$E\$, and each edge in \$E\$ is an ordered pair \$(u, v)\$, where \$u, v \in V\$.

The interconnect structure in an SOC circuit can thus be transformed into a graph \$G\$, and the vertex set includes all cores (terminal nodes) and fan-out nodes (intermediate nodes). A ring

Fig. 5. Interconnect diagnosis for  $n$ -bus structure.

$r$  is a subgraph  $r \subseteq G$  such that all the edges in  $r$  form a cycle. Since our goal is to diagnose the interconnect structure, we shall concern only the edges in the following discussion. Thus, a ring can be treated as a set of edges.

#### D. Problem Formulation

The goal of this paper is to find a set of test rings that achieve the optimal diagnosis resolution in the shortest time. To achieve this goal, we need to find a minimum number of diagnosis rings so that all faults can be correctly identified. The main difficulty, however, lies in the high complexity with the huge problem size. An SOC circuit usually contains a large number of interconnect wires, and the possible number of rings is likely to be exponential to the number of nets, although the exact number of rings depends on the interconnect structure. Consider the simple example, as shown in Fig. 5, in which  $m$  cores are connected by a bus of width  $n$ , denoted by  $n$ -bus. For simplicity, we shall assume that each core is passed by a ring only once. For example, the ring shown in Fig. 5 is of length 2.

In general, given an  $n$ -bus and a set of  $i$  modules, there are different ways to connect these modules into a ring of length  $i$ . Therefore, the total number of all possible rings in this system of  $m$  cores and an  $n$ -bus is

$$\sum_{i=2}^{\min(m,n)} C_i^m C_i^n.$$

Obviously, this number is at least exponential to  $\min(m, n)$ , and, thus, it is computationally intractable to search all possible rings and find a minimum subset of them for complete fault diagnosis. Even if we restrict the problem of the diagnosis check to a given set of rings, a brute-force exact algorithm is still very expensive. Therefore, it is desirable to find an efficient algorithm to achieve the optimal diagnosis resolution with a small number of test rings. The problem can be formally formulated as follows. We aim to develop an algorithm to find: 1) a small set of detection rings for 100% fault detection (IORT) and 2) an extra set of diagnosis rings for the optimal diagnosis resolution [interconnect OR diagnosis (IORD)]. Alternatively, if the optimal diagnosis resolution is not necessary, the algorithm should find the smallest set of test rings corresponding to the required resolution.

As defined earlier, a ring is a set of net segments forming a closed loop. However, the following constraints must be satisfied when a ring is generated.

- 1) **Hypernet Constraint:** If two edges  $e_i$  and  $e_j$  belong to the same hypernet and  $e_i$  is neither a downstream nor an upstream edge of  $e_j$ , they cannot belong to the same ring. In other words, if two edges  $e_i$  and  $e_j$  belong to the

same ring,  $e_i$  is either a downstream or an upstream of  $e_j$  (see Fig. 4).

- 2) **Frequency (Period) Constraint:** Let the wire delay of an edge  $e$  be  $d(e)$  and the delay of a wrapper cell  $w$  under an oscillation test be  $d(w)$ . The wire delay of an edge in a ring  $r$  can be detected if the following condition holds:

$$1/f_{\max} \leq \sum_{e \in r} d(e) + \sum_{w \in r} d(w) \leq 1/f_{\min}.$$

- 3) **Core constraint:** Let the total number of cores be  $m$  and the number of rings constructed at the same time in a test session  $T$  be  $|T|$ . We need at least one counter for each ring to measure whether the oscillation frequency is correct; therefore, at least  $|T|$  local counters are required in this session for delay measurement. Let the number of crosstalk fault detectable in this session be  $n_{\text{xtalk}}(T)$ . Since each target crosstalk fault must be checked by a local counter and each module is assumed to contain one local counter, the following condition holds:  $|T| + n_{\text{xtalk}}(T) \leq m$ .

Since the number of rings is usually too large to be checked exhaustively, it is difficult to find the minimum set of rings for fault detection and diagnosis. In order to handle the problem efficiently, we shall develop fast algorithms for diagnosability check and ring generation. In the remaining part of this paper, we discuss conditions that can facilitate diagnosability check for the given rings in Section III, and the ring generation algorithm is presented in Section IV. The test application time can be optimized further by applying tests concurrently or adaptively, and these techniques will be explored in Section V.

### III. INTERCONNECT DIAGNOSABILITY

In this section, we analyze interconnect diagnosability. We provide some diagnosability conditions for net segments (edges). With such conditions, we develop an algorithm for fast diagnosability check. We also derive the lower and upper bounds for the interconnect detection and diagnosis test scheme.

#### A. Diagnosability Analysis

Given a circuit consisting of  $n$  edges  $E = \{e_1, e_2, \dots, e_n\}$  and a set of  $m$  ORs  $R = \{r_1, r_2, \dots, r_m\}$ , once a ring is constructed, the test outcome is either “pass” (P) or “fail” (F). When an edge  $e_i$  is faulty, the test outcome of applying the  $m$  rings is said to be the syndrome of a faulty  $e_i$ .

*Definition 3:* A fault on edge  $e_i$  and a fault on edge  $e_j$  are distinguishable under the test set  $R$  if the syndrome of faulty  $e_i$  and faulty  $e_j$  are different.

*Definition 4:* An edge is said to be single-fault diagnosable under the test set  $R$  if a fault on the edge can be correctly identified, given that there is at most one fault in the interconnect structure.

Edge  $e_i$  is single-fault diagnosable if and only if its syndrome is different from all the other edges' syndromes. The diagnosability problem is to determine whether edge  $e_i$  is single-fault diagnosable under the test set  $R$ . Assume that edge  $e_i$  belongs

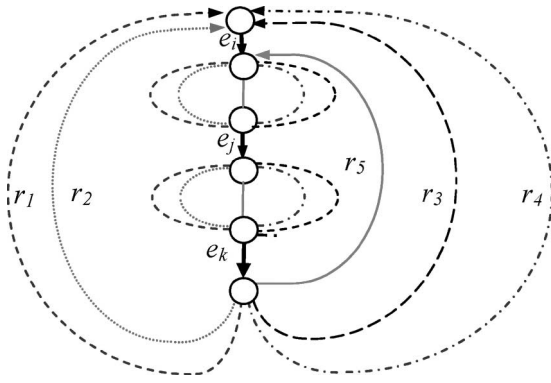


Fig. 6. Interconnect diagnosis graph example.

to a set of  $l$  different rings  $R_i = \{r | r \in R, e_i \in r\}$ . In other words,  $R_i$  is a subset of  $R$  with cardinality  $l$  ( $|R_i| = l$ ). Let  $E_i = \bigcap_{r \in R_i} r$  be the set of edges appearing in all rings of  $R_i$ , obviously,  $e_i \in E_i$ . An example is illustrated in Fig. 6, where  $e_i$  belongs to four different rings  $R_i = \{r_1, r_2, r_3, r_4\}$ , and thus,  $|R_i| = 4$ .  $E_i = r_1 \cap r_2 \cap r_3 \cap r_4 = \{e_i, e_j, e_k\}$  contains edges appearing in all rings in  $R_i$ .

**Lemma 1:** A fault on edge  $e_i$  and a fault on edge  $e_j$  are distinguishable under the test set.

*Proof:*  $\Leftarrow$  The fact  $R_i \neq R_j$  implies that there exists a ring  $r$  such that either: 1)  $r \in R_i \wedge r \notin R_j$  or 2)  $r \in R_j \wedge r \notin R_i$ . Thus, the syndromes of faulty  $e_i$  and faulty  $e_j$  are different.

$\Rightarrow$  When  $R_i = R_j$ , both faulty  $e_i$  and faulty  $e_j$  fail the same set of rings, and thus, they have the same syndrome. ■

**Theorem 2:** Edge  $e_i$  is single-fault diagnosable  $\Leftrightarrow R_i \neq R_j$  for all  $1 \leq j \leq n$  and  $j \neq i$ .

The correctness of Theorem 2 follows the result of Lemma 1. It takes  $\mathcal{O}(n^2m)$  time to verify Theorem 2 (where  $n$  is the number of nodes and  $m$  is the number of edges) since each pair of edges have to be compared. In order to reduce the complexity for diagnosability check, the following theorems can be used.

**Theorem 3:** Edge  $e_i$  is single-fault diagnosable if  $|E_i| = 1$ .

*Proof:* Assume that edge  $e_i$  is not single-fault diagnosable. From Theorem 2, there must exist an edge  $e_j$ , such that  $j \neq i$  and  $R_i = R_j$ . Therefore, both  $e_i$  and  $e_j$  belong to  $E_i$  and thus  $|E_i| > 1$ .

The application of Theorem 3 can greatly reduce the time complexity for the diagnosability check of an edge if the edge is single-fault diagnosable. However, the reverse of Theorem 3 is not true, since  $|E_i| = 1$  is only a sufficient condition for single-fault diagnosability.

Note that, when the sufficient condition given in Theorem 3 is true, we must have  $l \geq 2$ . When  $R_i$  has only one ring (i.e.,  $l = 1$ ),  $E_i$  is the set of all edges in this ring. Since a ring is consist of at least two edges,  $|E_i|$  must be greater than one.

When the intersection of  $l$  rings is consist of multiple edges, it is still possible to diagnose the faults as outlined in the following theorem.

**Theorem 4:** Let  $R'_i$  be any nonempty subset of  $R_i$  for an edge  $e_i$  and  $E'_i = \bigcap_{r \in R'_i} r$ . Edge  $e_i$  is single-fault diagnosable  $\Leftrightarrow \forall e_k \in E'_i - \{e_i\}$ , and  $e_i$  and  $e_k$  are distinguishable.

*Proof:*  $\Leftarrow$  When at least one ring in  $R'_i$  oscillates correctly,  $e_i$  must be fault-free. On the other hand, when no rings in  $R'_i$  os-

cillate correctly, at least one edge in  $E'_i$  is faulty. Since all edges in  $E'_i - \{e_i\}$  are distinguishable from  $e_i$ , we know whether  $e_i$  is faulty. Therefore,  $e_i$  is also single-fault diagnosable.

$\Rightarrow$  Assume that there is an  $e_k \in E'_i - \{e_i\}$  and  $e_k$  is not distinguishable from  $e_i$ . When every ring in  $R'_i$  fails, it may be attributed to either  $e_k$  or  $e_i$ . Thus,  $e_i$  is not single-fault diagnosable. ■

Theorem 4 shows that not all rings in  $R_i$  are necessary to diagnose  $e_i$ , and a subset  $R'_i$  is informative enough if and only if  $e_i$  is distinguishable with other edges in  $E'_i$ . The following corollary is a natural extension of Theorem 4.

**Corollary 5:** Let  $R'_i$  be any nonempty subset of  $R_i$  for an edge  $e_i$  and  $E'_i = \bigcap_{r \in R'_i} r$ . If for each  $e_k \in E'_i - \{e_i\}$ ,  $e_k$  is single-fault diagnosable, then edge  $e_i$  is also single-fault diagnosable.

An example for the above definitions, theorems, and corollaries is shown in Fig. 6. Let the edge under consideration be  $e_i$ ; then,  $R_i = \{r_1, r_2, r_3, r_4\}$ , and  $E_i = \{e_i, e_j, e_k\}$ . Since  $R'_i$  can be any nonempty subset of  $R_i$ , we may choose  $R'_i = \{r_2, r_3\}$ , and thus,  $E'_i = \{e_i, e_j, e_k\}$ . It is not necessary to have both  $e_j$  and  $e_k$  diagnosable to make  $e_i$  diagnosable. For example, let faults on  $e_j$  and  $e_k$  be indistinguishable; if a fault on  $e_i$  is distinguishable with  $\{e_j, e_k\}$ , then  $e_i$  is diagnosable according to Theorem 4.

Note that the above analysis applies to all types of faults except crosstalk glitches since they can be located directly from the test results of each ring. For example, consider the example shown in Fig. 1. If there are detectable glitches due to the crosstalk fault between wires  $a$  ( $a_1$  or  $a_2$ ) and  $b$  ( $b_1$  or  $b_2$ ), they will be observable through the counter in core  $C_3$  and  $C_4$ , and hence, the fault is located.

## B. Heuristic Diagnosability Check

In order to accelerate the process of diagnosability analysis, we propose a heuristic for diagnosability check in this section based on the definitions, lemmas, corollaries, and theorems developed in the previous theoretical analysis framework in Section III-A.

Consider two edges  $e_i$  and  $e_j$ . According to Lemma 1, faults on these two edges are distinguishable if  $|R_i| \neq |R_j|$ . Conversely, if faults on  $e_i$  and  $e_j$  are indistinguishable, then we must have  $|R_i| = |R_j|$ . Thus, as the first step, we sort and partition all edges according to the number of rings passing them (i.e.,  $|R_i|$  for edge  $e_i$ ). Edges  $e_i$  and  $e_j$  are put into the same group when  $|R_i| = |R_j|$ . Obviously, faults on two edges will be distinguishable if the two edges are in two different groups. Therefore, we only need to check whether the fault on an edge is distinguishable from faults on the edges that are in the same group as the target edge. The diagnosability analysis should start with the group with the highest  $|R_i|$ . For example, in Fig. 6,  $e_j$  and  $e_k$  are in the same group as  $|R_j| = |R_k| = 5$ , distinguishable from  $|R_i| = 4$ .

The second heuristic is to apply Theorem 3 first to check the diagnosability of an edge since it is much easier. Since the condition of Theorem 3  $|E_i| = 1$  is only sufficient but not necessary to guarantee that  $e_i$  be single-fault diagnosable, it is still possible that  $e_i$  is single-fault diagnosable when  $|E_i| \neq 1$ .

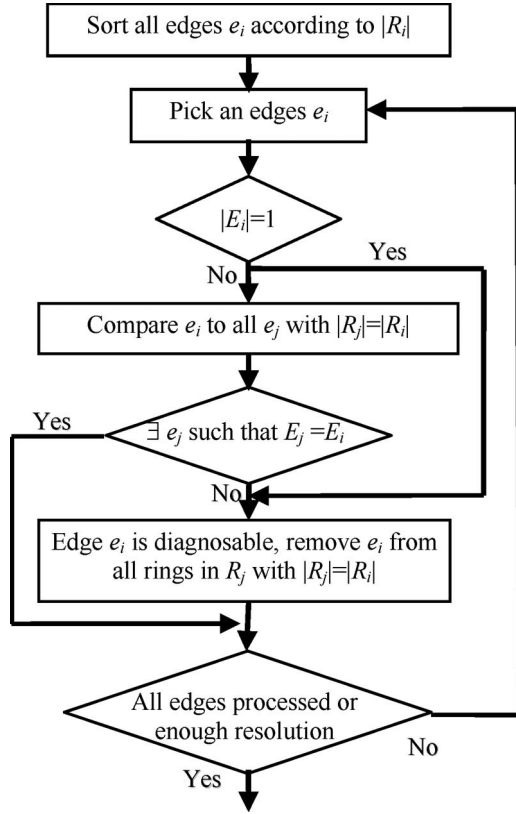


Fig. 7. Flow chart of heuristic for diagnosability checking.

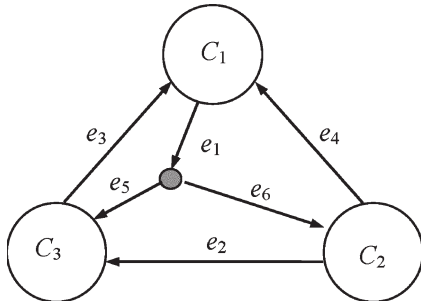


Fig. 8. Diagnosability example for Fig. 3(b).

In this case, we need to compare  $R_i$  with  $R_j$  for each  $e_j$  in the same group as  $e_i$ .

To avoid the aforementioned problem, a third heuristic is used. The most likely reason for diagnosable  $e_i$  with  $|E_i| \neq 1$  is that there exists an  $e_j$  such that  $R_j \supset R_i$ . When the edge  $e_j$  has been checked and removed from the checklist before edge  $e_i$  is processed, we shall not run into this problem. To further simplify the diagnosability check, whenever edge  $e_i$  is found to be single-fault diagnosable, it should be removed from all rings in  $R_i$ , as suggested by Corollary 5. The flowchart of the diagnosis checking heuristic is shown in Fig. 7.

Finally, when two faults are indistinguishable, they should be collapsed into the same equivalent class so as not to be compared twice.

The interconnect diagnosis heuristic algorithm is illustrated as follows. Consider the graph shown in Fig. 8, which is the graph model for Fig. 3(b). There are three rings in the

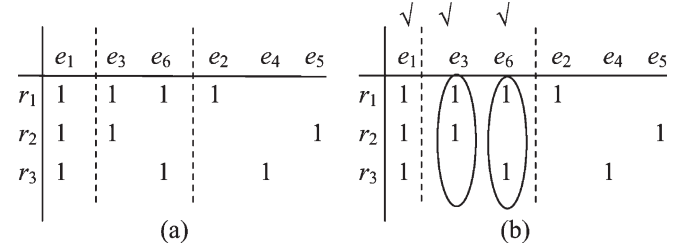


Fig. 9. Matrices for heuristic diagnosability checking.

figure:  $r_1 = \{e_1, e_2, e_3, e_6\}$  (ordered by  $e_1, e_6, e_2$ , and  $e_3$ ),  $r_2 = \{e_1, e_3, e_5\}$ , and  $r_3 = \{e_1, e_4, e_6\}$ .

A straightforward way to represent the diagnosability information is to use a matrix. The matrix representation for the example of Fig. 8 is illustrated in Fig. 9(a), where each column represents an edge and each row represents a ring. The entry  $(i, j)$  is “1” if ring  $i$  contains edge  $j$ . Note that the edges are sorted and partitioned into three groups that are separated by the dashed line. The first group consists of the edge  $e_1$ , which is contained in all three rings (i.e.,  $|R_1| = 3$ ). The second group consists of edges  $e_3$  and  $e_6$ , with each of them being contained in two rings (i.e.,  $|R_3| = |R_6| = 2$ ). The third group consists of the remaining three edges, with each of them being contained in only one ring (i.e.,  $|R_2| = |R_4| = |R_5| = 1$ ).

The syndrome of  $e_3 = \{110\}$  indicates that the test results of  $r_1$  and  $r_2$  are incorrect and  $r_3$  is correct when  $e_3$  is faulty; the syndrome of  $e_6 = \{101\}$  indicates that  $r_1$  and  $r_3$  are incorrect and  $r_2$  is correct when  $e_6$  is faulty. The diagnosability analysis is applied to the groups in the nonincreasing order of  $|R_i|$ . We start with the group with  $|R_i| = 3$  (i.e.,  $\{e_1\}$ ), followed by the group with  $|R_i| = 2$  ( $\{e_3, e_6\}$ ), and finally the group with  $|R_i| = 1$  ( $\{e_2, e_4, e_5\}$ ).

The diagnosability checking proceeds as follows. First, since the first group contains only one edge and the syndrome of  $e_1 = \{111\}$ ,  $e_1$  is single-fault diagnosable. Then, we check edges in the next group  $e_3$  and  $e_6$ . Edge  $e_3$  is contained in rings  $r_1$  and  $r_2$ , and the intersection of these two rings is  $\{e_1, e_3\}$ . Since edge  $e_1$  is diagnosable, edge  $e_3$  is single-fault diagnosable according to Corollary 5. Similarly, edge  $e_6$  is also diagnosable for the same reason.

Edges  $e_3$  and  $e_6$  are then marked and removed from the rings, as shown in Fig. 9(b). The reason is that they are already known to be diagnosable, which means they can be distinguished with any other faults. As a result, they do not need to be considered in the following process. There is only one edge remained in each ring, and thus, edges  $e_2, e_4$ , and  $e_5$  are single-fault diagnosable, again due to Corollary 5.

### C. Number of Tests

In this section, we analyze the lower and upper bounds on the test time or the number of tests in terms of the number of rings for our IORT scheme and IORD scheme.

The test time required for both detection and diagnosis is proportional to the number of rings. Therefore, it is important to minimize the number of rings required for either detection or diagnosis. In general, it requires more tests for fault location than fault detection.

For the IORT scheme, in an  $n$ -edge system, the lower bound on fault detection test is one if all the edges form a single large ring. This lower bound, however, is usually not achievable. A more realistic bound on fault detection tests is obtained by considering the pin order in cores. Thus, a smaller number of rings may be achievable through pin reordering. The upper bound of fault detection is  $n$ .

For the IORD scheme, to estimate the minimum number of tests required for diagnosis, we shall examine the theorems given in the previous section. In order to ensure that an edge is single-fault diagnosable regardless of other edges, it must belong to at least two rings, and it is the only common shared edge of these two or more rings, according to Theorem 3. A minimum set of rings satisfying this condition consists of  $\lfloor n/2 \rfloor$  distinct two-edge rings for the set of  $n$  edges. An illustrative example of this situation is shown in Figs. 8 and 9, for which edges  $e_1$  is single-fault diagnosable for  $|R_1| = 3$ ,  $e_3$  and  $e_6$  are diagnosable according to Theorem 3, and all the other edges are diagnosable according to Corollary 5.

Another interesting special case is the bidirectional bus. The bus lines in an  $n$ -bus (not including wires connecting cores to bus lines) can be diagnosed with  $n - 1$  rings, where a ring is constructed for every pair of adjacent nets. It can be verified that the internal  $n - 2$  lines are diagnosable due to Theorem 3, while the other two nets are diagnosable due to Corollary 5.

For a random interconnect structure, the number of diagnostic rings may be difficult to find. We estimate the number of rings required for diagnosis as follows. Let the number of rings required for fault detection be  $|R_t|$ . In the worst case, we need a new ring for each edge to satisfy Theorem 3, a total of  $m$  edges. Therefore,  $|R_d| = |R_t| + m$  predetermined rings should be enough for fault diagnosis if such rings exist. In general, if we can find a distinct ring for each net segment, we should be able to diagnose all net segments with  $m$  rings. These  $m$  rings are the extra test rings (or the number of tests) to achieve the optimal diagnosis resolution for each net segment (i.e., total  $|R_d|$ ), in addition to the original test time for the interconnect detection scheme ( $|R_t|$ ). We will show the details on how to get  $|R_d|$  and  $|R_t|$  in the following interconnect diagnosis algorithm in Section IV.

Also, we will show that the adaptive diagnosis can further reduce the number of tests from the predetermined approach of linear complexity to logarithmic complexity for a relatively balanced adaptive approach to be presented in Section V.

#### IV. INTERCONNECT DIAGNOSIS ALGORITHM

In this section, we integrate the fast diagnosability check heuristic developed in Section III and develop an efficient diagnosis ring generation algorithm. Before formally introducing our integrated IORD algorithm, we define important notations in the next paragraphs.

In order to uniquely identify the faulty net segment, we need to ensure the maximum diagnosability. The metric for interconnect diagnosability is the diagnosis resolution. The diagnosis resolution is defined as the maximum number of nets with the same syndrome under a given set of test rings. A higher resolution implies a smaller number of edges in each

<b>Algorithm:</b> IORT (Interconnect Oscillation Ring for Fault Detection)
<b>Input:</b> A hypergraph $H = (V, L)$ representing a circuit <b>Output:</b> A list of rings $R_i$
<ol style="list-style-type: none"> <li>1. Transform hypergraph <math>H</math> into a new graph <math>G = (V', E)</math> with equivalent 2-pin nets;</li> <li>2. <math>R_i = \emptyset</math>;</li> <li>3. <b>for every</b> <math>e = (u, v) \in E</math> and <math>e</math> is not visited</li> <li>4.     <math>R_i = R_i \cup \text{find\_ring}(G, e)</math>;</li> <li>5. reverse-order simulation for rings in <math>R_i</math>.</li> </ol> <b>function</b> find_ring( $G, e$ ) <ol style="list-style-type: none"> <li>1. Let <math>e = (u, v)</math> and <math>v</math> is an input pin in core <math>C</math>;</li> <li>2. <b>if</b> <math>v</math> is a pin in the starting core</li> <li>3.     <b>return</b> the ring and mark all nets as visited;</li> <li>4. <b>for every</b> output pin <math>w</math> in <math>C</math></li> <li>5.     <b>if</b> there is an unvisited edge <math>(w, x)</math></li> <li>6.         find_ring(<math>G, (w, x)</math>);</li> <li>7.     <b>else if</b> there is an untried output net <math>(w, x)</math></li> <li>8.         find_ring(<math>G, (w, x)</math>);</li> <li>9.     <b>else</b></li> <li>10.        <b>return</b> <math>\emptyset</math>;</li> <li>11. <b>end function</b></li> </ol>

Fig. 10. Ring generation for interconnect fault detection algorithm (IORT).

indistinguishable set. In general, we need more rings to achieve a higher level of diagnosis resolution. Our target is to diagnose every fault on every net segment, defined as the optimal diagnosis resolution.

We propose a heuristic to find a small set of rings for single-fault diagnosis. The SOC under test is modeled as a hypergraph  $H$ . This graph is then transformed into graph  $G = (V', E)$ , as outlined in Section II-D. The vertex set  $V'$  is consist of cores and fan-out points (intermediate nodes). The edge set  $E$  is consist of net segments partitioned from the original hypernets, as explained in Fig. 4(b). Our goal is to generate a predetermined set of rings to diagnose all edges in  $E$ .

Since we need to detect the interconnect structure before diagnosis, the set of fault-detection test rings  $R_t$  should be applied first. In order to find  $R_t$ , we propose a heuristic algorithm to find a minimum set of rings that cover all two-pin nets under test. The algorithm is a modified depth-first search and works as follows. The SOC under test is first modeled as a hypergraph  $H$  and then transformed into graph  $G = (V', E)$  with two-pin nets only. We generate a ring containing a two-pin net  $(u, v) \in E$  by starting from vertex  $v$ , an input pin. Then, we try to find an output pin  $w$  that locates in the same core as  $v$ , and  $w$  is connected to a two-pin net that is not yet covered by any other ring. If no such unvisited two-pin net  $(w, x)$  exists, we just select the first available output net from any available set of output pins. This process is repeated until a ring is found. The procedure then goes over again until all two-pin nets are covered.

The above heuristic works as follows. Whenever we start looking for a new ring, we explore the paths containing two-pin nets that are not yet covered. In this way, each new ring may cover as many other uncovered nets as possible. After all rings having been generated, a simple reverse order simulation is conducted to remove redundant rings. A net is OR testable if there exists at least one ring containing this net. The algorithm is shown in Fig. 10.



<p><b>Algorithm:</b> IORD (Interconnect Oscillation Ring Generation for Fault Diagnosis)</p>
<p><b>Input:</b> A hypergraph <math>H = (V, L)</math> representing a circuit</p> <p><b>Output:</b> A set of rings <math>R_d</math></p>
<ol style="list-style-type: none"> <li>1. Transform hypergraph <math>H</math> into a new graph <math>G = (V', E)</math> with equivalent 2-pin net segments;</li> <li>2. Generate a set of rings <math>R_t</math> for fault detection;</li> <li>3. <math>R_d = R_t</math>;</li> <li>4. Conduct diagnosability check;</li> <li>5. <b>for every</b> <math>e \in E</math> {</li> <li>6.   <b>if</b> (<math>e</math> is not single-fault diagnosable)</li> <li>7.     Find a ring <math>r</math> to make <math>e</math> diagnosable;</li> <li>8.     <math>R_d = R_d \cup \{r\}</math>;</li> <li>9.     Modify the diagnosability of all edges in <math>E</math>;</li> <li>10. } return <math>R_d</math>;</li> </ol>

Fig. 11. Ring generation for interconnect fault diagnosis algorithm (IORD).

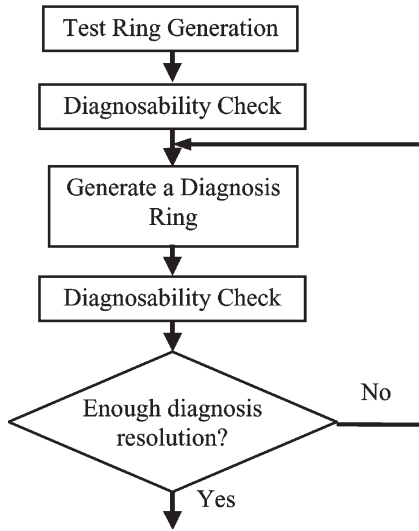


Fig. 12. Diagnosis ring generation procedure.

Our goal for the interconnect diagnosis is to find a small set of rings  $R_d$  that can uniquely identify the faulty edge or net segment if it exists. The set  $R_d$  is obtained by augmenting  $R_t$  as follows. We first apply the diagnosability checking techniques discussed in Section III to  $R_t$  to find out the net segments that are not diagnosable. For an edge  $e$  that is not single-fault diagnosable, we try to find a new ring passing it without going through the edges that are indistinguishable to  $e$ . If such a ring exists, it will be included in  $R_d$ . The diagnosability checking should be conducted for each added ring so that other edges that become diagnosable with the new ring will be found.

In this algorithm, we can achieve the highest diagnosis resolution when every net segment is diagnosable under  $R_d$ . With the reduced diagnosis resolution, the number of diagnosis rings can be reduced accordingly. Thus, this algorithm can be adjusted to the required diagnosis resolution to reduce the number of diagnostic rings. The algorithm for the generation of diagnosis rings is given in Fig. 11.

The flowchart illustrating the process of diagnosis ring generation is given in Fig. 12.

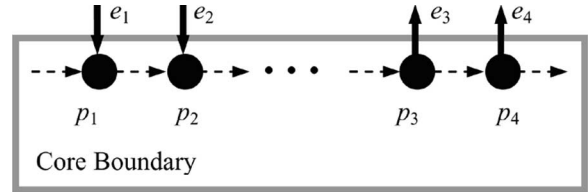


Fig. 13. Scan chain constraint.

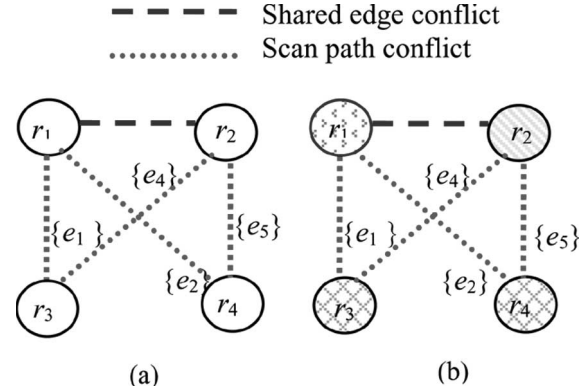


Fig. 14. (a) Conflict graph. (b) Graph coloring.

## V. OPTIMIZATION TECHNIQUES FOR INTERCONNECT DIAGNOSIS

The overall test and diagnosis time is determined by the number of test rings as well as how these rings are applied. In this section, we propose two optimization techniques for test time reduction: a concurrent test scheme and an adaptive diagnosis method.

### A. Concurrent Tests

Multiple ORs can be applied simultaneously as long as they do not interfere with each other. Two rings cannot be applied concurrently if they share some net segment or they go through the same scan path in a core. The condition is illustrated in Fig. 13 for scan path conflict. Assume that two rings with no common net segments pass the same core. The first ring contains edges  $e_1$  and  $e_3$ , while the second ring includes edges  $e_2$  and  $e_4$ . Although these two rings do not share common net segments, they cannot be applied at the same time due to the same scan path they go through.

In order to achieve the maximum concurrence or parallelism test, we model all the constraints by a conflict graph [25], as shown in Fig. 14(a). Each ring is represented by a node, and two nodes are connected by an edge if they interfere with each other for a scan-path constraint in Fig. 13 or a common-edge constraint. The problem of finding the maximum concurrence tests can thus be reduced to the well-known graph coloring problem [30], as shown in Fig. 14(b). Those rings/nodes colored with the same color can be tested concurrently, e.g.,  $r_3$  and  $r_4$  in Fig. 14(b), and we need at least three colors for the four nodes of Fig. 14. The coloring problem in the general graph has known to be NP-complete. (Nevertheless, in our experiment, we focus on the interconnect tree structure,

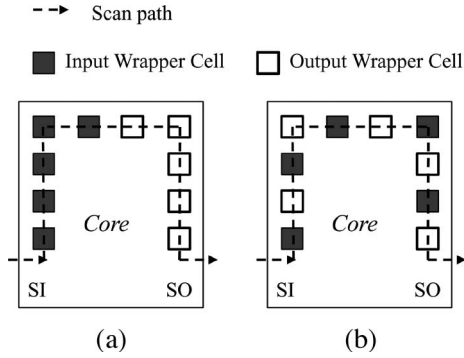


Fig. 15. Pin reordering for interleaving configuration.

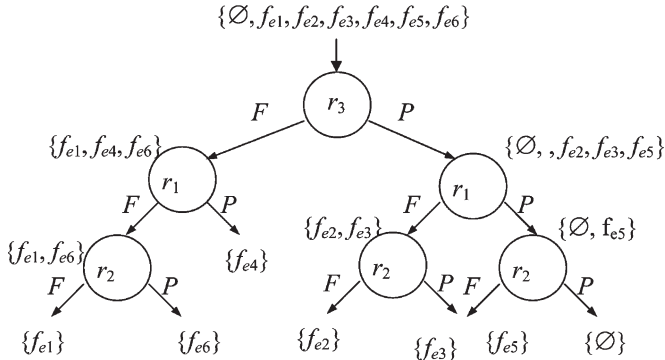


Fig. 16. Adaptive diagnosis tree.

for which the coloring problem can be solved in polynomial time [30].)

One possible way to reduce the number of “mutually exclusive” rings (i.e., rings that cannot be tested concurrently) is to reorder the pin positions. Consider the core illustrated in Fig. 15(a). The five nets connecting to the five input wrapper cells belong to different rings, and none of the rings can be tested at the same time due to the shared scan path constraint. However, if we reorder the pin positions such that input cells and output cells appear alternately, at most five rings can be formed simultaneously, with each ring passing two adjacent pins, as shown in Fig. 15(b).

### B. Adaptive Diagnosis

The number of test patterns can be greatly reduced whenever adaptive diagnosis is possible. In the adaptive diagnosis, a test pattern is selected according to the result of previous tests. An adaptive diagnosis tree, typically a binary tree, can be constructed according to the test patterns. For example, the adaptive diagnosis tree for the diagnosis example given in Figs. 8 and 9 is illustrated in Fig. 16.

For an  $n$ -net system, initially there are  $n + 1$  possible diagnosis results, namely fault-free ( $\emptyset$ ) and a single fault on net  $e_i$  ( $f_{ei}$ ) for  $1 \leq i \leq n$ . Each node in the tree represents a test pattern (ring), and the test outcome can be either pass (P) or fail (F). According to the test outcome of applying a ring, the indistinguishable set of edges can be divided into two groups. If the tree is balanced, the minimum number of diagnosis patterns required is  $\lceil \log_2(n + 1) \rceil$ .

In order to construct a balanced adaptive diagnosis tree, in each internal tree node, we need to select the test pattern (i.e., test ring) that evenly partitions the possible outcomes into two groups: fail (F) and pass (P). For example, in Fig. 16, we choose the test pattern  $r_3$  as the first test, since it evenly partitions the six possible outcomes into fail ( $f_{e1}$ ,  $f_{e4}$ , and  $f_{e6}$ ) and pass ( $\emptyset$ ,  $f_{e2}$ ,  $f_{e3}$ , and  $f_{e5}$ ). It can be seen that, in Fig. 16, each test partitions possible outcomes into two groups whose cardinalities differ by at most one in each level.

The upper bound on the number of adaptive diagnosis test sessions needed in our method can be computed as follows. Let the number of test rings (without diagnosis) be  $|R_t|$  and the length of the longest test ring be  $L_h$ . In the worst case, we need to apply  $|R_t|$  rings to find out that there is a faulty net, and the last ring contains  $L_h$  net segments which are all passed by the ring only. It takes up to  $L_h - 1$  rings to distinguish these  $L_h$  possible faults, and thus, the maximum number of diagnosis rings is  $|R_t| + L_h - 1$ .

## VI. EXPERIMENTAL RESULTS

We tested the diagnosis algorithm based on six commonly used MCNC benchmark circuits. The results are listed in Table I, where the first column gives the circuit names and the next four columns give the circuit statistics (“Statistics”), including the number of cores (#core), the number of pads (#pads), the number of hypernets (#hyp), and the number of net segments (#net\_seg). The 5th column, #net\_seg, lists the number of net segments, modeled as shown in Fig. 4(b), to be diagnosed in each benchmark. The next three columns (“Pre-determined”) give the experimental results for predetermined diagnosis, including the number of rings required to detect all two-pin nets ( $|R_t|$ ) and to diagnose all single faults ( $|R_d|$ ). In each benchmark, all net segments are single-fault diagnosable. The last column,  $|R_d|/|R_t|$ , gives the ratio of rings from 1.25X to 2.81X for the maximum diagnosis versus the rings for fault detection. This ratio means that we need extra test time of 1.25X to 2.81X to diagnose the single fault in each net segment under the predetermined diagnosis method compared to the IORT scheme.

In each case, we also give the estimated testing time (given in parenthesis), obtained by assuming only a 4-MHz measuring period, as discussed in Section II-A, to estimate the longest test application time for each ring. The time needed to set up the rings should be roughly proportional to the testing time.

Since in these benchmark circuits the net directions are not given, we make the following assumptions.

- 1) All cores are listed in a given order. For a hypernet, the pin corresponding to the first core in the core list is assumed the source, while the others are sinks.
- 2) Since the order on internal scan paths is not known, we conservatively assume that all output pins are placed in consecutive positions, and, thus, each ring may pass any core only once.
- 3) All I/O pads are connected through the boundary-scan path, while the positions of the pads are unknown, and, thus, the boundary-scan path appears only once in each ring.

TABLE I  
EXPERIMENTAL RESULTS FOR INTERCONNECT DIAGNOSIS BOTH FOR PREDETERMINED AND ADAPTIVE METHODS

Circuit	Statistics				Predetermined			Analysis				Adaptive		
	#core	#pad	#hyp	#net_seg	$ R_t $	$ R_d $	$ R_d / R_t $	#One Ring	#No Diag	#Equ Class	$ R_d  -  R_t $	max. EC	$ R_a $	$ R_d / R_a $
ac3	27	75	211	416	133(33.3ms)	374(93.5ms)	2.81	389	323	68	241	8	140(35ms)	2.67
ami33	33	42	117	343	242(60.5ms)	303(75.8ms)	1.25	309	126	59	61	5	246(61.5ms)	1.23
ami49	49	22	361	475	156(39ms)	386(96.5ms)	2.47	406	337	88	230	9	162(40.5ms)	2.38
apte	9	73	92	136	73(18.3ms)	122(30.5ms)	1.67	127	94	40	49	4	76(19ms)	1.61
hp	11	45	72	195	81(20.3ms)	164(41ms)	2.02	176	145	51	82	7	87(21.8ms)	1.89
xerox	10	2	161	356	218(54.5ms)	342(85.5ms)	1.57	346	214	86	124	5	222(55.5ms)	1.54
Comp.					0.9679								1	

TABLE II  
COMPARISON BETWEEN THEORETICAL BOUNDS AND EXPERIMENTAL RESULTS

Circuit	#NoDiag	#EquClass	Equation (3) (#NoDiag-#EquClass)	Extra Rings ( $ R_d  -  R_t $ )	(#NoDiag-#EquClass) and ( $ R_d  -  R_t $ )
ac3	323	68	255	241	14 (5.49%)
ami33	126	59	67	61	6 (8.96%)
ami49	337	88	249	230	19 (7.63%)
apte	94	40	50	49	1 (2.00%)
hp	145	51	94	82	12 (12.77%)
xerox	214	86	128	124	4 (3.13%)
Comparison			1.0712	1	6.64%

Assumption 2) makes concurrent test impossible. Thus, no concurrent tests are assumed in Table I. Assumption 3) makes the boundary-scan path appear only once in each ring. The previous three assumptions give the worst case scenario. The results are an upper bound on both test and diagnosis rings, and the actual number of rings should be smaller.

The next four columns (“Analysis”) give the diagnosis-related information after applying  $R_t$  rings. The column #OneRing gives the number of nets passed by only one ring. It can be seen that most nets are passed by one ring only when compared with the number of net segments (#net\_segment). Since the purpose of  $R_t$  is to detect faults with the minimum number of rings, it is not surprising that most nets are passed by one ring only. Most nets that are not diagnosable at this stage fall into this category. Columns “#NoDiag” and “#EquClass” give the number of nets that are not diagnosable and the number of equivalence classes after applying  $R_t$ , respectively, and they are the targets for further diagnosis. Two faults are in the same equivalence class if their syndromes for the tests are identical. The last column in this group (“ $|R_d| - |R_t|$ ”) gives the number of extra diagnosis rings required in each case to make all nets single-fault diagnosable. Note that, in an equivalence class of size  $s$ , we need no more than  $s - 1$  extra rings to distinguish these  $s$  nets. Assume that there are  $m$  equivalence classes whose sizes are  $s_1, s_2, \dots, s_m$ , respectively. The upper bound on the number of additional diagnosis rings “ $|R_d| - |R_t|$ ” can be expressed as follows:

$$\sum_{i=1}^m (S_i - 1) = \sum_{i=1}^m S_i - m = \#NoDiag - \#EquClass. \quad (3)$$

A comparison between theoretical bounds and experimental results is shown in Table II, in which the upper bound on the required number of extra rings ( $|R_d| - |R_t|$ ) is “(#NoDiag)-(#EquClass),” and it can be seen that these two numbers “(#NoDiag)-(#EquClass)” and “ $|R_d| - |R_t|$ ” are close in all

cases. Specifically, the empirical results “ $|R_d| - |R_t|$ ” differs from the theoretical results “(#NoDiag)-(#EquClass)” given in (3) by small differences of only up to 6.64%.

The last three columns in Table I (“Adaptive”) compare the number of rings required in both predetermined and adaptive diagnoses. The number of rings in a predetermined diagnosis is  $|R_d|$ . After applying  $R_t$  rings, the size of the largest equivalence class for each benchmark is given in the column “max. EC.” In the worst case, the adaptive diagnosis needs to apply  $|R_t|$  rings and then (max. EC)-1 rings for diagnosis. The number of the worst case adaptive diagnosis rings is given in column “ $|R_a|$ .” The last column ( $|R_d|/|R_a|$ ) shows the ratio of rings for the predetermined versus the adaptive diagnosis schemes. For the results shown in the column, the adaptive algorithm obtains 1.23X to 2.67X improvements over the predetermined diagnosis scheme. Also, from the normalized  $|R_a|$  and  $|R_t|$ , the test time of adaptive diagnosis is approximately equal to that for detection alone, which further reveals the effectiveness of adaptive diagnosis.

In summary, the OR scheme can detect and diagnose delay faults and crosstalk glitches very efficiently and effectively. In conventional schemes, the detection of crosstalk-induced glitches usually involves precise measurement of signals on the victim nets, for which complex clock control is needed for the delay fault detection due to the two-pattern tests. Therefore, more areas have to be devoted to the detection of errors due to these problems. In contrast, our scheme only slightly modifies IEEE 1500 wrapper cells, and the area overhead is small, as shown in Section II-A. Further, by applying the adaptive diagnosis technique, the time needed for diagnosis is approximately equal to that of detection alone. In other words, diagnosis can be accomplished with very small extra cost.

The experimental results for the concurrent test are given in Table III. The third column ( $|R_c|$ ) lists the number of test sessions after applying the concurrence test. When a set of rings are applied concurrently, we refer to these rings as a test

TABLE III  
CONCURRENT TEST SESSION

Circuit	$ R_d $	$ R_c $ (worst case)	$ R_d - R_c $
ac3	374	373	1 (0.27%)
ami33	303	290	17 (5.86%)
ami49	386	352	34 (9.66%)
apte	122	119	3 (2.52%)
hp	164	160	4 (2.50%)
xerox	342	327	15 (4.59%)
Comparison	1.0432	1	4.57%

session. The fourth column ( $|R_d| - |R_c|$ ) gives the percentage of improvements in the number of test sessions based on the worst case scenario of the interconnect structure. We note that the improvement can be even better for general interconnect structures. The reduction in test time due to the concurrent test ranges from 0.27% to 9.66% with no hardware overhead. Notice that the numbers give the lower bounds of empirical improvements by using the concurrence optimization technique.

The lack of concurrence is mainly a structure issue; however, it can also be attributed to several reasons. First, in the ring generation algorithm, we try to generate long rings so that the number of rings can be reduced. The longer rings tend to conflict with each other, and thus, they cannot be applied concurrently. Second, since we do not know the pin order in any core, we conservatively assume that each core can be passed by only one ring in a test session in order to avoid scan chain conflict. This may lead to an over pessimistic estimation on the scan path constraints and the number of test sessions. Third, the ring generation algorithm might not be perfect. The nets are searched according to their ordering in the data structure, and thus, some net segments are used more often than others, reducing the possibility of the concurrent test. Our future work should handle this problem.

The hardware overhead of the oscillation test scheme can be estimated as follows. Let the circuit size of a two-input NAND be an equivalent gate. In order to implement the proposed method, each wrapper cell must be enhanced to provide extra paths, as shown in Fig. 2. Besides, each core must be provided with an embedded counter. For an enhanced input wrapper cell, the area penalty is roughly 3.5 equivalent gates, which include two 2-to-1 multiplexers, one inverter, and a pulse detector [not shown in Fig. 2(a)]. For an enhanced output wrapper cell, the size of extra hardware is four equivalent gates, which include two 2-to-1 multiplexers, one buffer, and an inverting tri-state buffer.

A simple counter is constructed by cascading a number of T flip-flops (TFFs). There are many possible ways to design a TFF. For example, the design shown in [31] uses 5.5 equivalent gates to construct a resettable TFF. The length of the embedded counters is decided by the largest counter content  $n_{\max}$  since the counters must be large enough to accommodate  $n_{\max}$ . Thus, the size of the counter should be at least  $\lceil \log_2 n_{\max} \rceil$ . Since  $n_{\max} = f_{\max} \times T_0$ , by rewriting (2), we have

$$n_{\max} = f_{\max} \times T_0 \geq \frac{f_{\max}}{f_{\min}} \times \frac{1}{\zeta}. \quad (4)$$

From (4), we know that the length of the counters should be at least  $\lceil \log_2[(f_{\max}/f_{\min}) \times (1/\zeta)] \rceil$ .

TABLE IV  
HARDWARE OVERHEAD

Type	Enhanced Wrapper Cell		Embedded Counter
	input	output	
Area (in Equivalent gate counts)	3.5	4	$\left\lceil \log_2 \left( \frac{f_{\max}}{f_{\min}} \times \frac{1}{\zeta} \right) \right\rceil \times 5.5$

\* An equivalent gate stands for a two-input NAND gate.

In our experiment, we assume  $f_{\max} = 400$  MHz,  $f_{\min} = 4$  MHz, and  $\zeta = 0.001$ . Thus, the counter length should be 17, and the size of a counter is 93.5 equivalent gates. The area penalty is summarized in Table IV.

## VII. CONCLUDING REMARKS

In this paper, we have presented an IORD scheme for interconnect faults in SOC. In addition to the 100% fault detection coverage for each net achieved by the IORT scheme, we have shown that fault location or fault diagnosis can also be done by including some extra test rings to achieve the maximum diagnosis for each net segment. We have also presented two heuristics, diagnosability check and diagnosis ring generation, with theoretical study and integrated them into the IORD algorithm. We have further proposed the predetermined and adaptive diagnosis schemes and analyzed their costs. Finally, two optimization techniques for improving interconnect diagnosability are proposed and showed to be effective. Experimental results have justified the efficiency and effectiveness of the proposed methods.

## REFERENCES

- [1] M. Tehranipour, N. Ahmed, and M. Nourani, "Testing SoC interconnects for signal integrity using boundary scan," in *Proc. VTS*, 2003, p. 158.
- [2] Semiconductor Industry Association (SIA). [Online]. Available: <http://www.sia-online.org/>
- [3] International Technology Roadmap for Semiconductors (ITRS), (2004). [Online]. Available: <http://www.itrs.net/Links/2004Update/2004.htm>
- [4] W. K. Kautz, "Testing of faults in wiring interconnects," *IEEE Trans. Comput.*, vol. C-23, no. 4, pp. 358–363, Apr. 1974.
- [5] X.-T. Chen, F. J. Meyer, and F. Lombardi, "Structural diagnosis of interconnects by coloring," *ACM Trans. Des. Autom. Electron. Syst.*, vol. 3, no. 2, pp. 249–271, Apr. 1998.
- [6] Y. Kim, H.-D. Kim, and S. Kang, "A new maximal diagnosis algorithm for interconnect test," *IEEE Trans. Very Large Scale Integr.*, vol. 12, no. 5, pp. 532–537, May 2004.
- [7] C.-W. Yau and N. Jarwala, "A unified theory for designing optimal test generation and diagnosis algorithms for board interconnects," in *Proc. ITC*, 1989, pp. 71–77.
- [8] J.-C. Lien and M. A. Breuer, "Maximal diagnosis for wiring networks," in *Proc. Int Test Conf.*, 1991, pp. 71–77.
- [9] W.-T. Chen, J.-L. Lewandowski, and E. Wu, "Optima diagnostic methods for wiring interconnects," *IEEE Trans. Comput.-Aided Design*, vol. 11, no. 9, pp. 1161–1166, Sep. 1992.
- [10] E. J. Marinissen, B. Vermeulen, H. Hollmann, and R. G. Bennetts, "Minimizing pattern count for interconnect test under a ground bounce constraint," *IEEE Des. Test Comput.*, vol. 20, no. 2, pp. 8–18, Mar.–Apr. 2003.
- [11] M. Garey, D. Johnson, and H. Ho, "An application of graph coloring to printed circuit testing," *IEEE Trans. Circuits Syst.*, vol. CAS-23, no. 10, pp. 591–599, Oct. 1976.
- [12] W. Shi and W. K. Fuchs, "Optimal interconnect diagnosis of wiring networks," *IEEE Trans. Very Large Scale Integr.*, vol. 3, no. 3, pp. 430–436, Sep. 1995.
- [13] P. T. Wagner, "Interconnect testing with boundary scan," in *Proc. ITC*, 1987, pp. 52–57.
- [14] A. Krstic, L.-C. Wang, K.-T. Cheng, J.-J. Liou, and T. M. Mak, "Enhancing diagnosis resolution for delay defects based upon statistical timing and statistical fault models," in *Proc. DAC*, 2003, pp. 668–673.

[15] C. Su, Y.-T. Chen, M.-J. Huang, G.-N. Chen, and C.-L. Lee, "All digital built-in delay and crosstalk measurement for on-chip buses," in *Proc. DATE*, 2000, pp. 527–531.

[16] Y.-H. Choi and T. Jung, "System-level self-diagnosis in sparsely interconnected systems," *IEEE Trans. Rel.*, vol. 41, no. 3, pp. 433–439, Sep. 1992.

[17] M. Abramovici, C. Stroud, and M. Emmert, "Designing SOCs for yield improvement: Using embedded FPGAs for SOC yield improvement," in *Proc. DAC*, 2002, pp. 713–724.

[18] I. Harris and R. Tessier, "Diagnosis of interconnect faults in cluster-based FPGA," in *Proc. ICCAD*, 2000, pp. 472–475.

[19] V. Verma, S. Dutt, and V. Suthar, "Efficient on-line testing of FPGAs with provable diagnosabilities," in *Proc. DAC*, 2004, pp. 498–503.

[20] D. Chai, A. Kondratyev, Y. Ran, K. H. Tseng, Y. Watanabe, and M. Marek-Sadowska, "Novel design methodologies and signal integrity: Temporofunctional crosstalk noise analysis," in *Proc. DAC*, 2003, pp. 860–863.

[21] M. Cuvillo, S. Dey, X. Bai, and Y. Zhao, "Fault modeling and simulation for crosstalk in system-on-chip interconnects," in *Proc. ICCAD*, 1999, pp. 297–303.

[22] M. Nourani and A. Attarha, "Built-in self-test for signal integrity," in *Proc. DAC*, 2001, pp. 792–797.

[23] M. Kaneko and K. Sakaguchi, "Oscillation fault diagnosis for analog circuits based on boundary search with perturbation model," in *Proc. ISCAS*, 1994, pp. 93–96.

[24] K. Arabi and B. Kaminska, "Oscillation-based test strategy for analog and mixed-signal integrated circuits," in *Proc. VTS*, 1996, pp. 476–482.

[25] K. S.-M. Li, C.-L. Lee, C. Su, and J. E. Chen, "Oscillation ring based interconnect test for SOC," in *Proc. ASPDAC*, 2005, pp. 184–187.

[26] F. DaSilva, Y. Zorian, L. Whetsel, K. Arabi, and R. Kapur, "Overview of the IEEE P1500 standard," in *Proc. ITC*, 2003, pp. 988–997.

[27] D. Appello, P. Bernardi, M. Grosso, M. Rebaudengo, M. Sonza Reorda, and V. Tancorre, "On the automation of the test flow of complex SoCs," in *Proc. VTS*, 2006, pp. 166–171.

[28] B. I. Dervisoglu, "A unified DFT architecture for use with IEEE 1149.1 and VSIA/IEEE P1500 compliant test access controllers," in *Proc. DAC*, 2001, pp. 53–58.

[29] K. S.-M. Li, C.-L. Lee, C. Su, and J. E. Chen, "A unified approach to detecting crosstalk faults of interconnects in deep sub-micron VLSI," in *Proc. ATS*, Nov. 2004, pp. 145–151.

[30] M. R. Garey and D. S. Johnson, *Computer and Intractability: A Guide to the Theory of NP-Completeness*. San Francisco, CA: Freeman, 1979.

[31] N. H. E. Weste and K. Eshraghian, *Principles of CMOS VLSI Design: A Systems Perspective*, 2nd ed. Reading, MA: Addison-Wesley, 1992.

Dr. Su served as a Technical Program Committee (TPC) member for International Conference on Computer-Aided Design (ICCAD), Asia and South Pacific Design Automation Conference (ASP-DAC), Asian Test Symposium (ATS), and IEEE International Mixed-Signals Testing Workshop (IMSTW). He has also served as the Technical Program Cochair for ATS 2000 and General Cochair for ATS 2004. Presently, he also serves as the Executive Director for The National SOC Program, Taiwan.



**Yao-Wen Chang** (S'94–M'96) received the B.S. degree from National Taiwan University, Taipei, Taiwan, R.O.C., in 1988, and the M.S. and Ph.D. degrees from University of Texas, Austin, in 1993, and 1996, respectively, all in computer science.

He was with the IBM T.J. Watson Research Center, Yorktown Heights, NY, in the summer of 1994. From 1996 to 2001, he was with the faculty of National Chiao Tung University, Taiwan, R.O.C. Currently, he is a Professor with the Department of Electrical Engineering and the Graduate Institute of Electronics Engineering, National Taiwan University. He is also currently a Visiting Professor with the Waseda University, Japan. His current research interests lie in very large scale integration (VLSI) physical design, design for manufacturing, and FPGA.

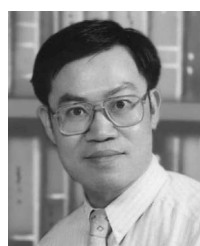
Dr. Chang received the Third Place at the 2006 ACM ISPD Placement Contest, the Best Paper Award at ICCD-1995 and six Best Paper Nominations from DAC-2005, 2004 ACM TODAES, ASP-DAC-2003, ICCAD-2002, ICCD-2001, and DAC-2K. He has received many awards for distinguished research, such as the 2005 and 2006 First-Class Principal Investigator Award and the 2004 Mr. Wu Ta You Memorial Award from National Science Council of Taiwan, and for excellent teaching from National Taiwan University and National Chiao Tung University. He currently serves on the technical program committees of a few important conferences on VLSI design automation and circuit design, including Asia Pacific Conference on Circuits and Systems (APCCAS), ASP-DAC, Design Automation and Test in Europe (DATE), ICCAD, International Conference on Computer Design (ICCD), International Symposium on Physical Design (ISPD), and VLSI-DAT. He is currently the chair of the Electronic Design Automation (EDA) Consortium of the Ministry of Education, Taiwan, R.O.C., a member of the board of governors of Taiwan IC Design Society, and a member of the IEEE Circuits and Systems Society, ACM, and ACM/SIGDA.



**Katherine Shu-Min Li** (S'04–M'06) received the B.S. degree from Rutgers University, New Brunswick, NJ, and the M.S. and Ph.D. degrees from National Chiao Tung University, Hsinchu, Taiwan, R.O.C., in 2001 and 2006, respectively.

She is currently an Assistant Professor with the Department of Computer Science and Engineering, National Sun Yat-sen University, Kaohsiung, Taiwan, R.O.C. Her research interests focus on crosstalk effects, signal integrity, systems-on-chip (SOC) testing, floor planning and routing for testability and yield enhancement, transition fault, scan reordering, scan routing, low-power scan techniques, especially on oscillation ring (OR) test scheme, and interconnect optimization in deep submicrometer and nanotechnology.

ability and yield enhancement, transition fault, scan reordering, scan routing, low-power scan techniques, especially on oscillation ring (OR) test scheme, and interconnect optimization in deep submicrometer and nanotechnology.



**Chauchin Su** (S'80–M'90) received the B.S. and M.S. degrees from National Chiao Tung University, Hsinchu, Taiwan, R.O.C., in 1979 and 1981, respectively, and the Ph.D. degree from University of Wisconsin, Madison, in 1990, all in electrical engineering.

In 1990, he joined the Department of Electrical Engineering, National Central University. In 2000, he transferred to the Department of Electrical and Control Engineering, National Chiao Tung University, Hsinchu, Taiwan, R.O.C., where presently he is

a Professor. His current research interests include the design and testing of high-speed serial links and mixed signal circuits. He is actively involved in various academic activities.



**Chung-Len Lee** (S'71–M'81) received the B.S. degree from National Taiwan University, Taiwan, R.O.C., in 1968, and the M.S. and Ph.D. degrees from Carnegie-Mellon University, Pittsburgh, PA, in 1971 and 1975, respectively, all in electrical engineering.

In 1975, he joined National Chiao Tung University as a faculty member, engaging in research on various topics on optoelectronics, semiconductor devices, integrated circuits, and processes. He has published more than 300 papers in the above areas.

Dr. Lee is a member of Phi Kappa Phi and Phi Tau Phi.



**Jwu E. Chen** (S'88–M'92) received the B.S., M.S., and Ph.D. degrees all in electronics engineering from National Chiao Tung University, Hsinchu, Taiwan, R.O.C., in 1984, 1986, and 1990, respectively.

He was a member of the Department of Electrical Engineering faculty at the Chung Hua University, Hsinchu, Taiwan, R.O.C., from 1990 to 2004. He is currently a member of the Department of Electrical Engineering faculty at the National Central University, Chungli, Taiwan. His research interests include multiple-valued logic, VLSI testing, reliable

computing, yield analysis, and humanoid robotics.